

# Bedingungen

## Inhaltsverzeichnis

- [1 Allgemeine Attribute](#)
- [2 ConditionTrue](#)
- [3 ConditionBetween](#)
- [4 ConditionBetweenOrEqualTo](#)
- [5 ConditionEqualTo](#)
- [6 ConditionGreaterThan](#)
- [7 ConditionGreaterThanOrEqualTo](#)
- [8 ConditionLessThan](#)
- [9 ConditionLessThanOrEqualTo](#)
- [10 ConditionIsEmpty](#)

Die Aktionen in der Kategorie Bedingungen

## 1 Allgemeine Attribute

Die allgemeinen Attribute **IgnoreError**, **Variable** und **Condition** können bei allen Aktionen angegeben werden. Die Attribute sind optional und brauchen nur bei Bedarf hinterlegt werden. Wenn diese für eine Aktion nicht benötigt werden, können diese aber auch zur besseren Lesbarkeit des Skriptes entfernt werden.

### IgnoreError

Das optionale Attribut **IgnoreError** gibt an, ob bei einem Fehler die Ausführung des Batchpad Skriptes abbricht oder das Skript weiter ausgeführt werden soll. Der Wert muss dem Typ Boolean (true oder false) entsprechen.

### Variable

Das optionale Attribut **Variable** kann immer dann verwendet werden, wenn man das Ergebnis einer auszuführenden Aktion ermitteln möchte. Variable="{ @ResultFileExists}".

Die Ergebnisse sind je nach ausgeführter Aktion vom Typ her unterschiedlich, oft ist es ein Boolean (true oder false) der angibt ob die Aktion erfolgreich war. Bei Aktionen für Zeichenketten sind die Ergebnisse dann eher vom Typ String usw.

### Condition

Das optionale Attribut **Condition** gibt an, ob die Aktion ausgeführt werden soll. Hierzu wird der Inhalt des Attributes als logischer Ausdruck auf Wahr oder Falsch geprüft. Der Ausdruck sollte dem Typ Boolean (true oder false) entsprechen.

Der Ausdruck kann Funktionen aus [VBScript](#) enthalten, genauso wie Operatoren NOT, OR, AND...

Mit dem Condition Attribut wertet man in der Regel Variablen aus, die Ergebnisse aus zuvor durchgeführten Aktionen enthalten. Beispiel: Condition="NOT { @ResultFileExists}"

## 2 ConditionTrue

Mit der Aktion **ConditionTrue** wird geprüft, ob die Bedingung in dem Attribut *Condition* zutrifft. Falls ja wird die in **ConditionTrue** verschachtelte Aktion ausgeführt.

Hinweis: Der Wert in dem Attribut *Condition* wird per VB-Skript Syntax ausgewertet.

```
<ConditionTrue Condition="" IgnoreError="false">
```

...

```
</ConditionTrue>
```

Beispiel für einen String-Vergleich in der Condition:

```
Condition="[*q]{@CurrentLanguage}[*q]=[*q]en[*q]"
```

## 3 ConditionBetween

Die Aktion **ConditionBetween** prüft, ob der Wert in dem Attribut *Value* zwischen den Werten in *MinValue* und *MaxValue* liegt. Das Ergebnis wird über das Attribut *Variable* zurückgegeben.

```
<ConditionBetween Value="" MinValue="" MaxValue="" Numeric="false" Not="false" Condition="" Variable="{@Result}" IgnoreError="false" />
```

## 4 ConditionBetweenOrEqualTo

Die Aktion **ConditionBetweenOrEqualTo** prüft, ob der Wert in dem Attribut *Value* zwischen den Werten *MinValue* und *MaxValue* liegt. Zusätzlich wird geprüft, ob der Wert *Value* dem Wert *MinValue* oder *MaxValue* entspricht. Das Ergebnis wird über das Attribut *Variable* zurückgegeben.

```
<ConditionBetweenOrEqualTo Value="" MinValue="" MaxValue="" Numeric="false" Not="false" Condition="" Variable="{@Result}" IgnoreError="false" />
```

## 5 ConditionEqualTo

Die Aktion **ConditionEqualTo** prüft, ob der Wert von dem Attribut *Value1* dem Wert von dem Attribut *Value2* entspricht. Das Ergebnis wird über das Attribut *Variable* zurückgegeben.

```
<ConditionEqualTo Value1="" Value2="" Numeric="false" Not="false" Condition="" Variable="{@Result}" IgnoreError="false" />
```

## 6 ConditionGreaterThan

Die Aktion **ConditionGreaterThan** prüft, ob der Wert von *Value1* größer als der Wert von *Value2* ist. Das Ergebnis wird über das Attribut *Variable* zurückgegeben.

```
<ConditionGreaterThan Value1="" Value2="" Numeric="false" Not="false" Condition="" Variable="{@Result}" IgnoreError="false" />
```

## 7 ConditionGreaterThanOrEqualTo

Die Aktion **ConditionGreaterThanOrEqualTo** prüft, ob der Wert von *Value1* größer oder gleich *Value2* ist. Das Ergebnis wird über das Attribut *Variable* zurückgegeben.

```
<ConditionGreaterThanOrEqualTo Value1="" Value2="" Numeric="false" Not="false" Condition="" Variable="{@Result}" IgnoreError="false" />
```

## 8 ConditionLessThan

Die Aktion **ConditionLessThan** prüft, ob der Wert von *Value1* kleiner als *Value2* ist. Das Ergebnis wird über das Attribut *Variable* zurückgegeben.

```
<ConditionLessThan Value1="" Value2="" Numeric="false" Not="false" Condition="" Variable="{@Result}" IgnoreError="false" />
```

## 9 ConditionLessThanOrEqualTo

Die Aktion **ConditionLessThanOrEqualTo** prüft, ob der Wert von *Value1* kleiner oder gleich *Value2* ist. Das Ergebnis wird über das Attribut *Variable* zurückgegeben.

```
<ConditionLessThanOrEqualTo Value1="" Value2="" Numeric="false" Not="false" Condition="" Variable="{@Result}" IgnoreError="false" />
```

## 10 ConditionIsEmpty

Die Aktion **ConditionIsEmpty** prüft, ob der Wert von *Value* leer bzw. nicht definiert ist. Das Ergebnis wird über das Attribut *Variable* zurückgegeben.

```
<ConditionIsEmpty Value="" Variable="{@Result}" />
```